Event handling using static and dynamic task allocation strategies

Brice Platerrier, Martin Vassor and Arnaud Wald

Abstract— **To optimize division of labor amongst several agents for different types of tasks with limited knowledge of the environment, we designed and tested different variations of the threshold-based algorithm. We show that even if non-communicating agents achieve relatively good performance, adding communication between agents to modify the decision process improves the efficiency and the robustness of the results.**

I. INTRODUCTION

This project deals with task allocation where there are multiple types of tasks. The robots will handle tasks that appear throughout the environment according to different task allocation strategies.

The main task allocation strategies can be divided into two categories: market-based and threshold-based approaches. A comparative study of both task allocation strategies is presented in [3]. Though market-based task allocation is more efficient when information is accurate, the threshold-based approaches show similar results at a lower computational cost when the knowledge of the environment and the tasks is more limited. In our peculiar case where robot vision and communication capabilities are low, threshold-based approaches are more suited and will be studied through the course of this project.

As exposed in [2], the thresholds can either be considered fixed or variable according to various adaptation models. Moreover, the decision can either be deterministic or probabilistic (e.g. using a sigmoid function). Another possibility is to endow the robots with communication capabilities so that they can calibrate their thresholds based on received information from their neighbors [1].

In this project, we first propose static strategies where the weights of the stimuli are fixed and the thresholds are individual. We compare the homogeneous experiment, where all robots have the same threshold, to the heterogeneous case, where the robots adapt there thresholds depending on the time spent in search or by specializing in a peculiar task.

In an attempt to improve performance by avoiding multiple robots selecting the same task, we analyze the benefits of a dynamic strategy where each robot emits a virtual potential field for the task type it selected. The robots use the intensities of the potential field separately for each type of task and use this additional information to dynamically adjust the weights of the stimuli.

II. EXPERIMENTS

To assess the efficiency of our algorithms, we consider the task handling problem where 3 types of tasks appear in a closed arena. A task, represented as a colored cylinder (with colors being red, green or blue), is processed by a robot being in its vicinity for a given amount of time, but is not processed faster as the number of robots close to its position increases. Once a task is processed, it disappears and a new one appears at a random location. The goal of our algorithms is to optimize the task allocation so as to prevent multiple robots from picking the same task.

Robots identify the tasks with the on-board camera. Each robot evaluates stimuli for red, green and blue tasks based on its local camera observations. Each robot analyzes the row of pixels in the middle of its camera image, then computes the sizes of the biggest clusters of each color. The initial stimuli for red, green and blue are those sizes in pixel, but they can be modulated to allow a better repartition of robots among tasks. The robot will then select the task according to our threshold-based algorithms.

In order to emit the virtual potential field, the robots are endowed with radio emitters (and receivers) that broadcasts (and gathers) the relevant information perceived by the robots, within a short range. These information are then fed into our public and individual threshold-based algorithm to dynamically adapt the stimuli.

Implementation: The framework used during the course of the project allowed us to test 3 main approaches. They could be divided in two categories, one being private (static case) and the other being public (dynamic case). In the first case, we considered both homogenous (fixed and identical thresholds among robots) and heterogeneous (adaptive thresholds based on each robot's vision or time spent in search mode) cases. In the second case, we took advantage of the local potential field emissions perceived by the robots to adjust the weights of each color stimulus to improve task allocation and tried both fixed and variable thresholds. Simulations were run on the realistic robot simulator WeBots. The agents used are e-pucks robots. The arena configuration can be seen on figure 1.

Task Allocation Mechanism: For the simulations to be comparable, we used the same Finite State Machine (FSM) for all simulations and all types of algorithms. This FSM is described in Figure 2. The behavior of the robots could be described as follows. All the robots are initialized in the first state ($S1$) that consists in spinning around and processing the image from the camera to determine if task needs to be handled. Once a task is picked, the robot changes state and goes directly towards the task ($S2$). To avoid collisions between robots, obstacle avoidance is added is this state. If the robot happens to lose track of its target (when the stimulus gets lower than a given threshold), it switches back to $S1$. When the robot is close enough to the task, it slows down for a given amount of time to process the task ($S3$).

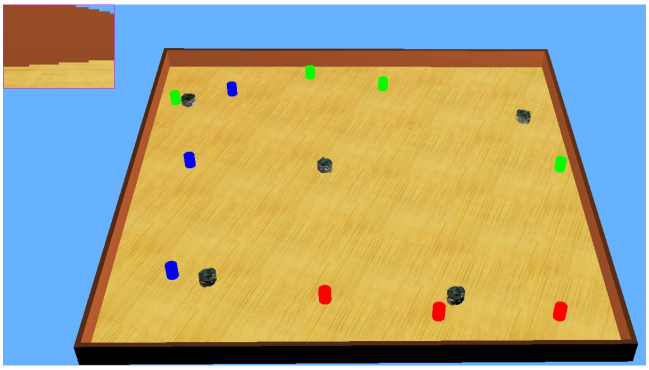

Fig. 1. The disposition of the arena with the e-puck robots (in gray/dark green) and the different tasks represented by red, green and blue cylinders. On the top-left corner is displayed the camera image of one of the robots.

Once the task is processed, the robot will go back in $S1$.

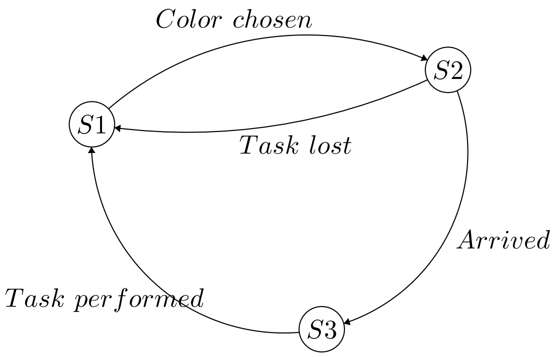

Fig. 2. The Finite-State-Machine of the robots

A. Private, Fixed Threshold-based Algorithm

In this first experiment, the same response threshold is assigned to each robot and identical for each color (no specialization). Various agent behaviors are obtained thanks to the local perception of the environment and the private assessment of the demand. We considered both the deterministic and probabilistic task allocation. In the latter, the robots where using a sigmoid function to determine if a given task should be processed or not.

B. Private, Adaptive Threshold-based Algorithm

The next case considers adaptive thresholds to improve the robustness of the task allocation strategy. As the number of robots is not equal nor a multiple of the number of task categories, we chose to consider two adaptation methods. The first one considers the robots to be "colorblind", the threshold is the same for each color and the adaptation consists in lowering the value of all the thresholds as the robots remain in search mode (and conversely, increasing the value if a task is performed). The second one proceeds to a specialization of the robots as they perform a given type of task repeatedly. This approach can be combined with the previous one by lowering the thresholds of the robots if too much time is spent in search mode.

C. Public, Fixed & Variable Threshold-based Algorithm

The final approach adds communication between the robots to share information about the tasks identified through vision. The method consists in using the information received through radio communication to change the weights of the color stimuli so as to avoid selecting the same task as another robot.

In order to neglect the influence of neighbors being too far from a robot position, we chose to limit the maximum range at which the emissions were perceived. The value was set so that the corresponding area of reception of a given robot was one fifth of the total area of the arena. Thus, each robot was more inclined to work in its own area and could deal with neighbors entering the area and likely to choose the same task. Furthermore, we chose to modify the saturate the signal strength of the received emissions. Indeed, for the robot to react sufficiently fast if it picked the same task as its neighbor, the signal strength had to be limited to a maximum value before normalization of the received perceptions. This assumption somehow goes beyond the initial idea of potential field but is perfectly workable using radio communications. It should also be noted that the model adopted here could be applicable to really noisy and imprecise receiver.

The sent information had to be meaningful for the robots to assess the situation correctly. Although we tried to send the proximity of the closest task for each color, we eventually restrained the information sent to the closeness of the chosen task. This information was then updated and sent as long as the robot was traveling to the task. Based on this information, the robots could decide to search for a new task if the same color was chosen by a neighbor that was closer to the task.

In this project, we propose a function to update each color stimulus based on the information perceived by the neighbors. This function is comprised of two factors: the weight of local stimuli, α, and the weight of neighbors stimuli, β. The first factor reduces the power of the stimuli as the received values increases. The second one allows the robot to still have a chance to perform the task it had chosen if it is closer to it than its neighbors. If the decrease in the stimulus is high enough (i.e. the stimulus is less than a given threshold, called "lost threshold"), the robot will abandon the task it had chosen and resume search. The update of the stimulus for color i is done as follows:

$$\sigma_{new}(i) = \sigma_{prev}(i) * (1 + \beta) - \alpha * R(i) \quad (1)$$

where σ_{new} and σ_{prev} are the new and previous stimulus values and $R(i)$ is the normalized reception of color i.

III. Results & Discussion

In this section, we present and compare the results gathered from all experiments at sub-microscopic modeling level. Each task allocation run, for a team of 5 robots placed in a 160X160 arena, lasted 3 minutes (simulated time) and 40 runs were carried out for each simulation setup. The colored bars stand for the average performance and the error bars represent the standard deviations among runs.

Different parameters were tuned for each case scenario and are described in the subsequent subsections. Furthermore, unless otherwise stated, our stimuli were the number of pixels (from the processed image) corresponding to the closest task (cylinder) for each color.

A. Performance Metrics

The purpose of this project is to assess the benefit of dynamic, or adaptive, thresholds-based algorithms over static ones (whether it be the homogeneous or heterogeneous case). To do so, we introduce two performance metrics:

- The number of tasks processed within a time period (N/T)
- One minus the distance traveled per processed task $(1 - d/N)$

Since it is better to travel less per processed task, the transformation above was necessary so that all agents want to maximize those two metrics. While the former gives us an idea of the average rate at which actions are performed, the second embeds two notions at the same time: the distribution of workload among robots and the efficiency of the allocation in terms of distance from robots to tasks, which can be useful on real robots where the power supply is limited.

B. Private Fixed Threshold-based Task Allocation

For these experiments, we compare the performances obtained for two different fixed thresholds in both deterministic and probabilistic cases. In the probabilistic case, a sigmoid function - computed from the threshold θ and the stimulus σ - is used to proceed to the task allocation.

$$F(\sigma) = \sigma^n/(\theta^n + \sigma^n) \tag{2}$$

Figure 3 and Figure 4 shows the results obtained through the simulation of a private fixed-threshold task allocation scheme. As it can be seen, in the case of fixed thresholds and deterministic responses, higher values of thresholds usually mean that less task will be performed and yield really poor performances. Consequently, only a few robots are processing the tasks and thus travel larger distances for a given number of task (see (a) and (b)). In the probabilistic case, the robots are more or less inclined to choose tasks that are further away. A sufficiently low steepness allows for the larger threshold to be compensated (see (c) and (d)). However if the steepness of our sigmoid function gets too high, the performances plummet and are comparable to the deterministic case (see (e) and (f)).

C. Private Variable Threshold-based Task Allocation

The fixed-threshold approach does not allow the robots to travel far away from their latest position. Consequently, the task-allocation showed poor performances when the tasks were massively spawned on one side of the arena. Hence, we tried to endow the robots with the capability of adapting their threshold based on the time they spend searching for a task to handle. At each time step spent in search mode, the threshold value is decremented by δ_θ. Each time a task is processed, the threshold is incremented by 1. As this function is computed at each time step (i.e. every 64 ms), its value must be low so that the adaptation is not too fast (long distances will be traveled) or too slow (time will be wasted in search mode).

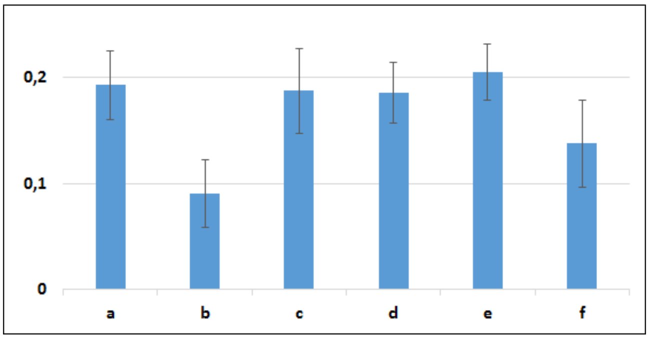

Fig. 3. Number of events handled per time unit for the private, fixed-thresholds algorithms with different thresholds θ and steepnesses n. (a) $\theta = 5, n = +\infty$, (b) $\theta = 8, n = +\infty$, (c) $\theta = 5, n = 10$, (d) $\theta = 8, n = 10$, (e) $\theta = 5, n = 30$, (f) $\theta = 8, n = 30$.

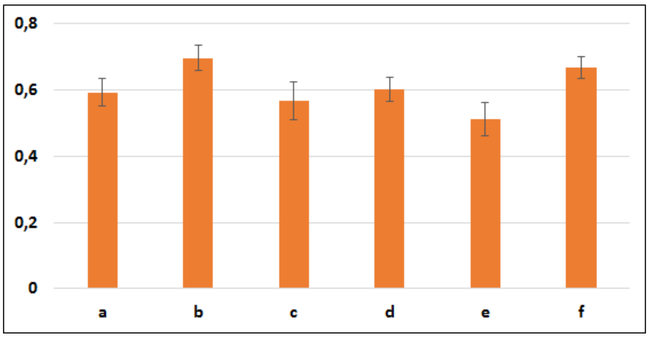

Fig. 4. Second metric for the private, fixed-thresholds algorithms with different thresholds θ and steepnesses n. (a) $\theta = 5, n = +\infty$, (b) $\theta = 8, n = +\infty$, (c) $\theta = 5, n = 10$, (d) $\theta = 8, n = 10$, (e) $\theta = 5, n = 30$, (f) $\theta = 8, n = 30$.

Although the poor quality and the limited field of view of the robot's vision allows for some noise (heterogeneity) and randomness to be added to the process, they do not prevent the robot from choosing the same task. Since the tasks were categorized, we tried to add specialization mechanism to thwart this tendency. Each time a task of a given category was performed, its corresponding threshold was decreased by a specified value and the other thresholds were increased by a different value.

The results are shown in Figure 5, Figure 6, Figure 7 and Figure 8. From Figure 5 and Figure 6, we can infer that the number of task processed per time unit is not significantly influenced by the speed of the adaptation process. On the flip side, the second performance metric suffers from the unreasonably low threshold and show that a larger distance was traveled by the robots. From Figure 7 and Figure 8, we can conclude that the specialization process is not really adapted to our case study. Indeed, the specialization implies that a higher distance is traveled by the robots on average and the number of tasks handled per time unit greatly decreases. This phenomenon is emphasized due to the number of robots being greater than the number of types of tasks. In an attempt to improve the performances, we also tried to mix the first adaptive approach with the specialization mechanism. As it can be seen in Figure7 and Figure 8, the number of task handled is then significantly higher but the second performance metric results still show that a large distance was traveled. In the remainder of the project, the specialization mechanism was abandoned in favor of the first, and only the first, method.

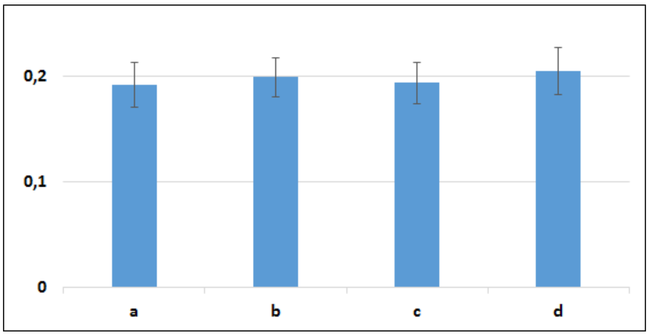

Fig. 5. Number of events handled per time unit for the private, adaptive-thresholds algorithms with a steepness $n = 30$ and different thresholds θ and adaptation factors δ_θ. (a) $\delta_\theta = 0.1$, (b) $\delta_\theta = 0.01$, (c) $\delta_\theta = 0.1$(d) $\delta_\theta = 0.01$.

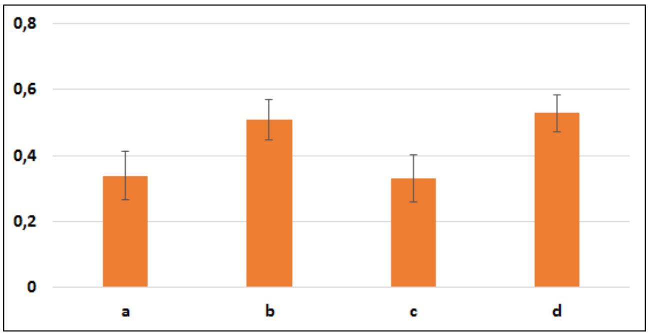

Fig. 6. Second metric for the private, adaptive-thresholds algorithms with a steepness $n = 30$ and different thresholds θ and adaptation factors δ_θ. (a) $\delta_\theta = 0.1$, (b) $\delta_\theta = 0.01$, (c) $\delta_\theta = 0.1$(d) $\delta_\theta = 0.01$.

D. *Public Fixed & Variable Threshold-based Task Allocation*

In order to allow the robots to drop a chosen task if a neighbor is likely to be heading towards it at the same time, we endowed the robots with communication capabilities in the form of potential fields emitted through the radio emitter/receiver. This potential field conveyed the stimulus corresponding to the chosen task color. With this additional information the robots were able to adapt their stimuli, according to equation (1).

We tested both adaptive and static setups to show the efficiency of the first adaptation approach in the public case. The parameters used were the ones who gave the best results thus far (i.e. $\theta = 5$, $\delta_\theta = 0.01$). Furthermore, as the probabilistic case did not yield significant improvements over the deterministic in the public case, we only present the results in the latter scenario.

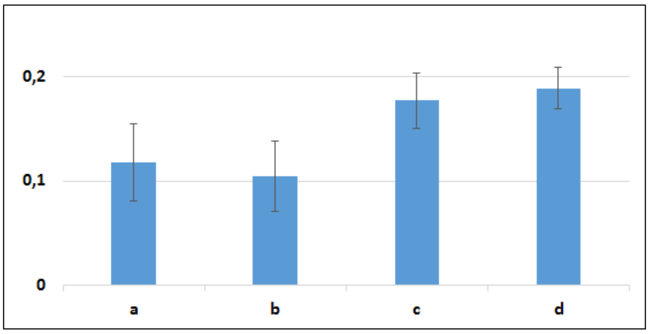

Fig. 7. Number of events handled per time unit for the private, adaptive-thresholds with specialization algorithms with a steepness $n = 30$, an initial threshold of $\theta = 5$ and different specialization rates. (a) Specialization rate $-1/+1$ and no adaptation, (b) Specialization rate $-1/+2$ and no adaptation, (c) Specialization rate -0.5/+0.5 and adaptation $\delta_\theta = 0.01$, (d) Specialization rate -1/+1 and adaptation $\delta_\theta = 0.01$.

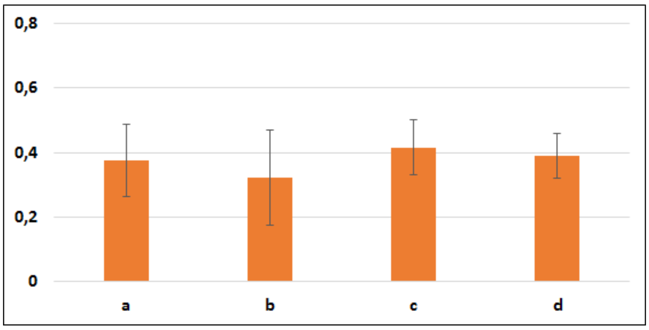

Fig. 8. Second metric for the private, adaptive-thresholds with specialization algorithms with a steepness $n = 30$, an initial threshold of $\theta = 5$ and different specialization rates. (a) Specialization rate $-1/+1$ and no adaptation, (b) Specialization rate $-1/+2$ and no adaptation, (c) Specialization rate -0.5/+0.5 and adaptation $\delta_\theta = 0.01$, (d) Specialization rate -1/+1 and adaptation $\delta_\theta = 0.01$.

The performances are shown in Figure 9 and Figure 10. Unsurprisingly, the adaptation mechanism allows for less inconsistencies in the results (lower variance) and a higher number of task to be processed per time unit (though it slightly increases the total traveled distance). More importantly, the communication between robots yields to better task allocation compared to the private case with adaptive thresholds. Finally, the public variable-threshold based approach shows better consistency than the private fixed threshold approach and overall better performance (lower variance, higher number of processed task, slightly higher distance traveled). Therefore, this last approach comforts us in the benefits of using communication in the form of potential fields to improve the task allocation through dynamic tuning of the stimuli.

IV. Conclusion

With this project, we have presented three scalable, distributed and threshold-based approaches for task allocation in both static and dynamic cases with limited knowledge about the tasks and demand. We compared the efficiency and robustness of each approach in our case study involving 5 robots and 3 types of task in a large arena.

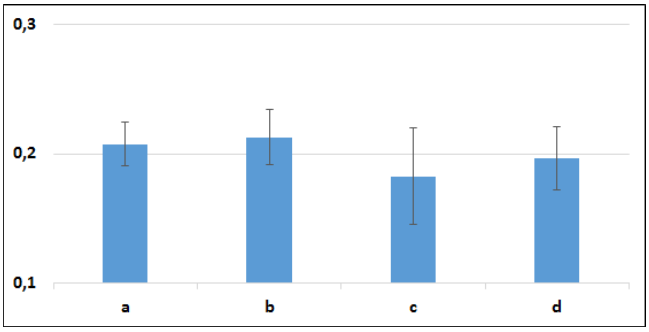

Fig. 9. Number of events handled per time unit for the public, adaptive-thresholds algorithms with $\alpha = 1$, $n = \infty$, a communication range of 40cm and different values of β. (a) $\beta = 0.1$ and adaptation $\delta_\theta = 0.01$, (b) $\beta = 0.2$ and adaptation $\delta_\theta = 0.01$, (c) $\beta = 0.1$ and no adaptation, (d) $\beta = 0.2$ and no adaptation.

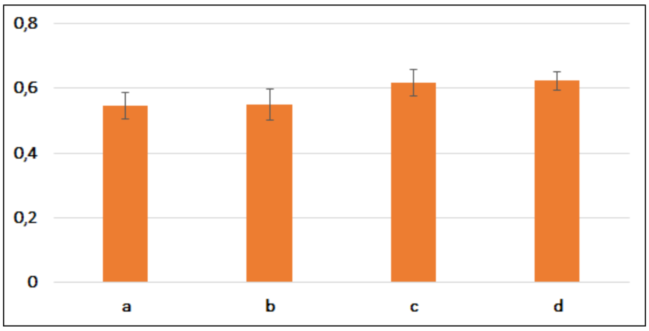

Fig. 10. Second metric for the public, adaptive-thresholds algorithms with $\alpha = 1$, $n = \infty$, a communication range of 40cm and different values of β. (a) $\beta = 0.1$ and adaptation $\delta_\theta = 0.01$, (b) $\beta = 0.2$ and adaptation $\delta_\theta = 0.01$, (c) $\beta = 0.1$ and no adaptation, (d) $\beta = 0.2$ and no adaptation.

Robots controlled using probabilistic threshold based algorithms showed no real improvements over the deterministic case scenario, even for high steepness. The specialization of robots turned out not to be adapted to our case study as the number of robots is greater than the number of task categories and the different types of tasks are not equally represented at each time step (in both time and space). On the contrary, adaptation mechanism lowering the thresholds as the time in search for a task increases proved its efficiency in improving the robustness (and thus consistency) of the task allocation. Finally, we showed the benefits of the addition of communication to improve the task allocation mechanism. This allowed the robots to reconsider their choice of task to handle while heading towards it, hence preventing two robots from choosing the same task.

However, it should be noted that our study would greatly benefit from the implementing of an optimization algorithm such as Particle Swarm Optimization (PSO). Unfortunately, we were not able to implement it in time. We also believe that adding more randomness into the process would yield better overall performance. This could be done through the addition of a random time for spinning in search mode or random walk when the robots are too close (as suggested in [1]). Finally, it could be interesting to study the consequences of allocating more time to search state for the robots to achieve a complete spin, and adapting their threshold based on what the perceived (vision, local perception) in a similar manner as [1].

References

[1] William Agassounon and Alcherio Martinoli. "Efficiency and robustness of threshold-based distributed allocation algorithms in multi-agent systems". In: *Proceedings of the first international joint conference on Autonomous agents and multi-agent systems: part 3*. ACM, 2002, pp. 1090–1097.

[2] Bonabeau et al. "Division of Labor and Task Allocation". In: *Swarm Intelligence*. 1999, pp. 91 – 109 –147.

[3] Nidhi Kalra and Alcherio Martinoli. "Comparative study of market-based and threshold-based task allocation". In: *Distributed Autonomous Robotic Systems 7*. Springer, 2006, pp. 91–101.